Networking of the Future: Software Defined Network: Day 8
Today’s Class

• Drawbacks of current Networking Paradigms
  – Motivation for SDN

• SDN!!!!!!

• Whitebox Networking

• OpenFlow: A common SDN API

• SDN challenges and Use-cases
HOW BIG IS THE SDN MARKET?

$35 BILLION BY 2018

2013: $1.5B
2014: $3.4B
2015: $7.8B
2016: $14.8B
2017: $24.4B
2018: $35.6B
VMware Buys Nicira For $1.26 Billion And Gives More Clues About Cloud Strategy

Posted Jul 23, 2012 by Alex Williams (@alexwilliams)
AT&T: Sorry vendors, SDN is eating your lunch

The cold winds of disruption start to whistle through telco-tin-land
Facebook just fired another big shot at Cisco — and dissed it a little, too
Networking Today:

- Distributed, time-consuming and error prone
  – Think BGP, Distance-Vector
Networking Today:

- Distributed, time-consuming and error prone
  – Think BGP, Distance-Vector

\[
\begin{align*}
128.35.8.*/24 & \quad 128.35.6.*/24 \\
128.35.7.*/24 & \quad 128.35.9.*/24
\end{align*}
\]
Ideally…

- Managing network in a *simple* way
- *Directly* and *explicitly* apply policies to network
Instead ...

- Managing network in a complex way
- No clear idea of the consequences

How can I change distance vector?
Is iBGP running in this network?
Should I worry about spanning-tree?

Forwarding tables
Change weights
How do you change BGP/ISP?

- Router configuration files
  - Low level commands
  - Think assembly

Specify link costs
*must be the same on both sides of a link

```plaintext
!configures a link
Interface vlan901
ip address 10.1.1.5 255.0.0.0
ospf cost 100

!configures a routing protocol
Router ospf 1
router-id 10.1.2.23
network 10.0.0.0 0.255.255.255
```
## Evolution of Network Provisioning: 1996-2013

### 1996

```
Router> enable
Router# configure terminal
Router(config)# enable secret cisco
Router(config)# ip route 0.0.0.0 0.0.0.0 20.2.2.3
Router(config)# interface ethernet0
Router(config-if)# ip address 10.1.1.1 255.0.0.0
Router(config-if)# no shutdown
Router(config-if)# exit
Router(config)# interface serial0
Router(config-if)# ip address 20.2.2.2 255.0.0.0
Router(config-if)# no shutdown
Router(config-if)# exit
Router(config)# router rip
Router(config-router)# network 10.0.0.0
Router(config-router)# network 20.0.0.0
Router(config-router)# exit
Router(config)# exit
Router# copy running-config startup-config
Router# disable
Router>
```

**Terminal Protocol:** Telnet

### 2014

```
Router> enable
Router# configure terminal
Router(config)# enable secret cisco
Router(config)# ip route 0.0.0.0 0.0.0.0 20.2.2.3
Router(config)# interface ethernet0
Router(config-if)# ip address 10.1.1.1 255.0.0.0
Router(config-if)# no shutdown
Router(config-if)# exit
Router(config)# interface serial0
Router(config-if)# ip address 20.2.2.2 255.0.0.0
Router(config-if)# no shutdown
Router(config-if)# exit
Router(config)# router rip
Router(config-router)# network 10.0.0.0
Router(config-router)# network 20.0.0.0
Router(config-router)# exit
Router(config)# exit
Router# copy running-config startup-config
Router# disable
Router>
```

**Terminal Protocol:** SSH
The End Results?
Microsoft pins Azure outage on configuration error

Outage caused by a 'safety valve' mechanism meant to prevent cascading network failures
Can We make things Simple?
Provide direct control?
Why don’t we have direct control?

- Networking today: Vertical integrated stacks
  - Similar to PC in 1980s (or phones in the early 2000s)
  - No choice on interface
  - Stuck with proprietary interfaces (even if bad!)
Implications on Networking...

• Restricted to ill defined vendor CLI

• Limited innovation

• Lots of Bugs!!!
  – Huge operating costs (OpEx)

• Monopolies
  – Huge Mark-ups (CapEx)
Software Defined Networking

- SDN decouples the control algorithms form the hardware
  - Introduces a nice API for communicating directly with the switches.

- Switch Operating System: exposes switch hardware primitives
IBM to OEM Brocade Switches. It's IBM vs HP.
WHAT’S INSIDE A SWITCH?

Application

Network OS

Hardware

Driver

Box

Silicon

Commodity parts!!!!
HP, Dell brand OEM
Why Can we have a nice API?

**Layer 3: (Distance vector)**
1. Matches on IP address
2. Forwards on interface (link)

**Layer 2.5: (VLAN)**
1. Matches on VLAN
2. Floods the packet

**Layer 2: (Spanning Tree)**
1. Matches on MAC address
2. Forwards on a port
   OR
2. Floods the packet

All switches match on same part of packets and perform same action.

**HP Magic Protocols**
- RIP
- VLAN
- SPT

**Cisco Magic Protocols**
- RIP
- VLAN
- SPT

**Juniper Magic Protocols**
- RIP
- VLAN
- SPT
Implications of SDN

Current Networking

SDN Enabled Environment

Distance Vector
Network O.S.
ASIC

Distance Vector
Network O.S.
ASIC

Distance Vector
Network O.S.
ASIC

Distance Vector++

Global View

Controller (N. O.S.)

Programmatic Control

Southbound API

Switch O.S
Switch HW

Switch O.S
Switch HW

Switch O.S
Switch HW
Implications Of SDN

Current Networking

- Distributed protocols
  - Each switch has a brain
  - Hard to achieve optimal solution
- Network configured indirectly
  - Configure protocols
  - Hope protocols converge

SDN Enabled Environment

- Global view of the network
  - Applications can achieve optimal
- Southbound API gives fine grained control over switch
  - Network configured directly
  - Allows automation
  - Allows definition of new interfaces
• **Southbound API**: decouples the switch hardware from control function
  - Data plane from control plane

• **Switch Operating System**: exposes switch hardware primitives
ONUG Board & Members Include ...

- Fidelity
- Bloomberg
- Bank of America
- JPMorgan Chase
- Gap Inc
- Citi
- UBS
- FedEx
- Cigna
- Credit Suisse
- Pfizer
Section 2: WhiteBox Networking
Networking with Linux

- Applications
- Controller (Network O.S.)
- Southbound API
- Switch Hardware

SDN
Edge-Core AS5610-52X (with ONIE)

from $5,095.00

Edge-Core AS6701-32X (with ONIE)

from $7,895.00
Section 3: Southbound API: OpenFlow
OpenFlow

- Developed in Stanford
  - Standardized by Open Networking Foundation (ONF)
  - Current Version 1.4
    - Version implemented by switch vendors: 1.3

- Allows control of underlay + overlay
  - Overlay switches: OpenVSwitch/Indigo-light
How SDN Works: OpenFlow

- Applications
- Controller (N. O.S.)
- Switch O.S.
- Southbound API

OpenFlow connections:
- Between Applications and Controller
- Between Controller and Switch O.S.
OpenFlow: Anatomy of a Flow Table Entry

1. Forward packet to zero or more ports
2. Encapsulate and forward to controller
3. Send to normal processing pipeline
4. Modify Fields

Match: Switch Port, VLAN ID, VLAN pcp, MAC src, MAC dst, Eth type
Action: IP Src, IP Dst, IP ToS, IP Prot, L4 sport, L4 dport
Counter: VLAN pcp, IP ToS, Priority
Time-out: When to delete the entry
What order to process the rule
# of Packet/Bytes processed by the rule
OpenFlow: Types of Messages

- **Asynchronous (Controller-to-Switch)**
  - Send-packet: to send packet out of a specific port on a switch
  - Flow-mod: to add/delete/modify flows in the flow table
  - Read-state: to collect statistics about flow table, ports and individual flows
  - Features: sent by controller when a switch connects to find out the features supported by a switch
  - Configuration: to set and query configuration parameters in the switch

- **Asynchronous (initiated by the switch)**
  - Packet-in: for all packets that do not have a matching rule, this event is sent to controller
  - Flow-removed: whenever a flow rule expires, the controller is sent a flow-removed message
  - Port-status: whenever a port configuration or state changes, a message is sent to controller
  - Error: error messages

- **Symmetric (can be sent in either direction without solicitation)**
  - Hello: at connection startup
  - Echo: to indicate latency, bandwidth or liveliness of a controller-switch connection
  - Vendor: for extensions (that can be included in later OpenFlow versions)
Section 4: SDN Use Cases + Challenges
SDN Use Cases

- Network Virtualization (VMWare, Azure)
- Port tapping (Big Switch’s BigTap)
- Access control (Big Switch’s SNAC)
- WAN Traffic Engineering (Google B4)
- DDoS Detection (Defense4All)
- Network Orchestration (OpenStack, VMWare)
SDN Use Cases

• Network Virtualization (VMWare, Azure)
• Port tapping (Big Switch’s BigTap)
  – Cheaper more flexible version of Cisco’s port span
  – Cisco’s port span: limited to 2 per switch.
  – BigSwitch: limited to the number of ports on the switch
• Access control (Big Switch’s SNAC)
• WAN Traffic Engineering (Google B4)
  – New functionality
  – Drive network utilization up 80-90%
  – Huge cost savings
  – Dynamically adjust to failures
• DDoS Detection (Defense4All)
• Network Orchestration (OpenStack, VMWare)
SDN Use Cases

- WAN-Traffic engineering
  - Google’s B4 (SIGCOMM 2013)
  - Microsoft’s SWAN (SIGCOMM 2013)
- Network Function Virtualization: Service Chaining
  - SIMPLIFY/FlowTags (SIGCOMM 2013, NSDI 2014)
  - Slick (ONS 2013)
- Network virtualization
  - Nicira, Azure, Google,
  - VL2 & Portland (SIGCOMM 2009)
  - CloudNaaS (SoCC 2011)
- Seamless workload (VM) mobility
  - (CrossRoads (NOMS 2012))
- Data Center Traffic engineering
  - Routing elephant flows differently (Hedera – NSDI 2010)
  - Routing predictable traffic (MicroTE – CoNext 2011)
- Port-Mirroring
  - BigTap
  - OpenSafe (INM/WREN 2011)
Section 5: Challenges
Controller Availability
Controller Availability
Controller Availability

“control a large force like a small force: divide and conquer”

--Sun Tzu, Art of war

- How many controllers?
- How do you assign switches to controllers?
- More importantly: which assignment reduces processing time
- How to ensure consistency between controllers
SDN Reliability/Fault Tolerance

Existing network survives failures or bugs in code for any one device.

Controller: Single point of control
- Bug in controller takes the whole network down.
SDN Reliability/Fault Tolerance

Existing network survives failures or bugs in code for any one device.

Controller: Single point of control
- Bug in controller takes the whole network down
- Single point of failure
SDN Security

If one device in the current networks are compromised the network may still be safe

Controller: Single point of control
• Compromise controller
SDN Security

- Compromise controller
- Denial of Service attack the control channel
Data-Plane Limitations

• Limited Number of TCAM entries
  – How to fit network in limited entries?

• Limited control channel capacity
  – Need to rate limit control messages

• Limited switch CPU
  – Limit control messages and actions that use CPU
Conclusion

- SDN \(\rightarrow\) Ongoing Networking paradigm shift
  - Decouple Control from Data
  - Motivating factors:
    - Lower cost: CapEx & OpEx
    - New functionality: e.g. Google’s B4

- Whitebox Networking: commodity switches
  - Whitebox = bare-hardware + O.S.
  - Bare-metal = hardware.